

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) EP 1 022 860 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
26.07.2000 Bulletin 2000/30

(51) Int. Cl.⁷: H03M 13/29

(21) Application number: 00101174.1

~~This Page Blank (uspto)~~

(22) Date of filing: 21.01.2000

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States
AL LT LV MK NO SI

(72) Inventor:
Srinivasa Somayazulu, V.
OR 97223 (US)

(74) Representative:
MÜLLER & HOFFMANN Patentanwälte
Innere Wiener Strasse 17
81667 München (DE)

(30) Priority 22.01.1999 US 116765 P
28.07.1999 US 363303

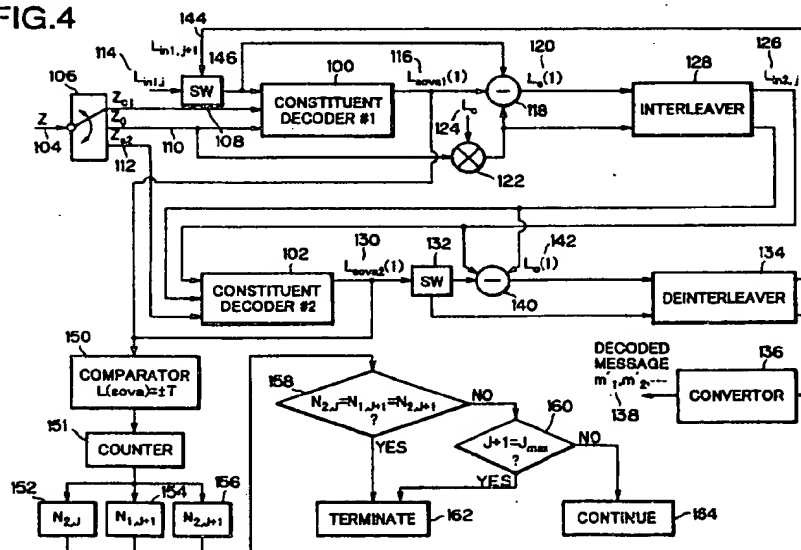
(71) Applicant
Sharp Kabushiki Kaisha
Osaka-shi Osaka (JP)

(54) Turbo decoding with variable number of iterations

(57) Turbo codes used in digital communication systems are decoded by successive iterations of a maximum likelihood algorithm. The turbo code decoder is a modular apparatus with each module comprising two or more serial constituent decoders (100, 102). The output of a constituent decoder (100) or a module is part of the input to the next constituent decoder (102) or module, respectively. The progress of the decoding can be monitored by establishing a limit for the output of the constituent decoders (100, 102) and monitoring the numbers of outputs of one or more constituent decoders (100,

102) that approximately equaling that limit. When the numbers of outputs approaching the limit or saturating is unchanged for successive iterations or successive serial decoders (100, 102) no further progress is being made in decoding the message. The performance of the turbo code decoder can be optimized by terminating iterations of the decoding process when the numbers of constituent decoder outputs that have saturated is unchanged for successive iterations or decoder operations.

FIG.4



EP 1 022 860 A1

Description

BACKGROUND OF THE INVENTION

[0001] The present invention relates to channel coding for digital communications systems and, more particularly, to a turbo code decoder useful for channel coding and a method for optimizing the performance of a turbo code decoder.

[0002] Forward error correction (FEC) is a system of error control for data transmission systems where the receiver is capable of detecting and correcting errors in the "as received" message induced by noise in the transmission channel. FEC is useful in connection with data transmission systems which lack a reverse channel with which retransmission of data can be requested or where retransmission would be difficult because the delay would be excessive or repeated retransmission would be required because of the number of expected errors. For these reasons, FEC has been of particular interest and use in wireless communication and space probe and satellite data transmission systems. FEC relies on channel coding where input message sequences are mapped to code symbol sequences that add redundancy and memory to the data before transmission.

[0003] Generally, channel coding utilizes block or convolutional coding to add redundancy to the message bit stream. Block coding breaks the bit stream representing the message into fixed size blocks and independently adds redundant code symbols to each block. Block coding is usually decoded with algebraic techniques. On the other hand, convolutional coding continuously adds redundant symbols to the bit stream based on the contents of the stream. In the convolutional encoder, the bits of the message are shifted serially into and out of a shift register having a number of individual registers. The output code is the result of modulo arithmetic performed on the contents of the shift register and, in some cases, the input bit stream as each successive message symbol or bit is shifted into the register. While bit stream segmentation is not required for convolutional coding, the coded bit stream is typically broken into blocks or frames for other reasons before transmission. Decoding of convolutional codes is accomplished with a heuristic or trial-and-error approach.

[0004] Turbo codes are produced by encoders comprising two, or more, parallel constituent encoders. The constituent encoders are often, but not necessarily, identical convolutional encoders. An interleaver or permuter is attached to the input of one or more of the constituent encoders. The interleaver rearranges the input of the attached constituent encoder in a systematic, pseudo-random manner. As a result, turbo codes comprise two or more (depending on the number of encoders) independently coded symbol streams that refer to the same input information bit stream. Turbo codes are of particular interest because with a relatively simple constituent code and large interleaver their performance can be near the theoretical or Shannon limit of the transmission channels.

[0005] Turbo code decoding is an iterative process with the results of a first modular decoder forming part of the input to a second modular decoder and so forth until the required number of iterations is achieved. When the turbo code is composed of two parallel concatenated codes, the modular turbo code decoder comprises two serially connected constituent decoders separated by an interleaver that reorders the output of the first decoder so that it may be used as input to the next decoder. Decoders for turbo codes with more than two parallel constituent codes may take a number of forms. A convolutional encoder is a state machine that codes by tracing a path through a code tree or trellis on the basis of the sequence of input symbols. From the symbols of the "as received" coded message the convolutional code decoder attempts to retrace the encoder's path through the code tree or trellis outputting the symbols of the decoded message while correcting errors incurred in transmission. One technique for decoding convolutional codes relies on algorithms which retrace the path of "maximum likelihood" through the trellis. One such "maximum likelihood" algorithm used in turbo code decoding is the soft output Viterbi algorithm (SOVA). A constituent decoder applying the SOVA algorithm computes or estimates the "log likelihood ratio," the logarithm of the ratio of the conditional probabilities of receiving the two outcomes (binary "1" and "0") given the observed signal value. The output of the constituent decoder is a plurality of signed numbers. The sign expresses the polarity of the decoded message symbol. The magnitude is a "soft" or analog value expressing the probability that the decoded symbol is the same as the original message symbol.

[0006] Generally, the turbo code decoder converges on a final decoded symbol sequence with successive iterations and the error rate performance improves until a threshold number of iterations is reached. While the error rate performance of the decoder generally improves with additional iterations, the rate of improvement decreases. Each iteration takes time further delaying completion of decoding. Heretofore, the number of iterations to be performed by a particular turbo code decoder was hard wired into the decoder. Optimizing the number of iterations to be hardwired into the decoder involves compromises in the error rate and latency of the decoder's performance. Further, due to the random nature of noise, "as received" data sequences are unequally corrupted and require different numbers of iterations to achieve the same level of error correction.

[0007] What is desired, therefore, is a turbo code decoder and a method of decoding that optimize the performance of the decoder producing a given level of error correction in the fewest number of decoding iterations on the average. Further, it is desired that the operation of the decoder be responsive to the error correcting requirements of each mes-

sage. Optimizing the decoding process reduces the latency in decoding a message at an acceptable error rate and reduces the cost and complexity of the decoder.

SUMMARY OF THE INVENTION

[0008] The present invention overcomes the aforementioned drawbacks of the prior art by providing a method of optimizing the performance of an iterating turbo code decoder having a plurality of constituent decoders comprising the steps of establishing a limit for an output of a constituent decoder, determining the number of decoder outputs approximately equaling the limit for each of successive iterations by the turbo code decoder, and terminating operation of the turbo code decoder when the numbers of the decoder outputs approximately equaling the limit are substantially unchanged for successive iterations. The progress of a turbo code decoder in decoding a message sequence can be monitored by establishing a limit for the output of a constituent decoder and monitoring the number of outputs of equaling or approaching the limit. If the number of outputs approaching the limit or saturating does not change for successive iterations by the turbo code decoder, no progress is being made in decoding the message. Operation of the turbo code decoder can be terminated without loss of data.

[0009] A second technique applying the method of optimizing the performance of an iterating turbo code decoder including a plurality of serial constituent decoders comprises establishing a limit for an output of a constituent decoder, determining a first and a second number of the outputs approximately equaling the limit produced by the second serial constituent decoder while performing a first iteration and a second iteration, respectively; determining a third number of the outputs approximately equaling the limit produced by the first serial constituent decoder while performing the second iteration; and terminating operation of the turbo code decoder when the first, second, and third numbers of outputs are substantially equal.

[0010] A third technique for applying the method of optimizing the performance of an iterating turbo code decoder including a plurality of serial constituent decoders comprises establishing a limit for an output of a constituent decoder; determining a first number of outputs produced by the first serial constituent decoder performing an iteration approximately equaling the limit; determining a second number of outputs produced by the last serial constituent decoder performing the iteration approximately equaling the limit; and terminating operation of the turbo code decoder when the first and second numbers of outputs are substantially equal.

[0011] A fourth technique of applying the method of optimizing the performance of an iterating turbo code decoder including a plurality of serial constituent decoders comprises establishing a limit for an output of a constituent decoder; and terminating operation of the turbo code decoder when substantially all outputs produced by the constituent decoder while performing an iteration are approximately equal to the limit. By monitoring the progress of the decoding process and terminating decoding iterations when progress slows or stops, the delay in decoding any particular message stream can be minimized while achieving a satisfactory error rate for any "as received" symbol sequence.

[0012] An optimized iterating turbo code decoder is also provided comprising a plurality of serial constituent decoders; a comparator to compare a plurality of outputs of a constituent decoder to a threshold value for the output; a counter to count the number of the outputs produced by the constituent decoder approximately equaling the threshold value; and a decision unit to terminate operation of the turbo code decoder when the number of the outputs approximately equaling the threshold value is unchanged for successive iterations. In a second embodiment of the optimized turbo code decoder, the number of outputs of the first and last of the serial constituent decoders that have saturated during an iteration is compared to determine progress in decoding. In a third embodiment, the number of outputs of the first serial constituent decoder performing a first iteration and the numbers of outputs of the last serial constituent decoder performing the first and a subsequent iteration are compared to determine decoding progress and whether further iterations are warranted.

[0013] The foregoing and other objectives, features and advantages of the invention will be more readily understood upon consideration of the following detailed description of the invention, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 is a block diagram of a communication link incorporating channel coding. FIG. 2 is a block diagram of an exemplary systematic, recursive convolutional encoder. FIG. 3 is a trellis diagram of the operation of a recursive, constituent convolutional encoder of the encoder of FIG. 2. FIG. 4 is a block diagram of a turbo code decoder according to the present invention. FIG. 5 is a trellis diagram for a soft output Viterbi algorithm-based, constituent convolutional decoder of the turbo code decoder of FIG. 4.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0015] Referring to FIG. 1, forward error correction (FEC) is used to detect and correct errors which can occur in digital messages as a result of noise in a communication link 2. A message 4 comprising a sequence of bits or symbols (m_1, m_2, \dots, m_n) originates with an information source 6. If the communication link 2 incorporates FEC, the message sequence 4 is encoded before transmission to a code symbol sequence 8, (C_1, C_2, \dots, C_n) by an encoder 10. The code-word sequence 8 is forwarded to a modulator 12 where it is converted to signals 14 ($\{s_i(t)\}$) which are suitable for the transmission in the channel 16 of the communication link 2. The transmission channel 16 may be affected by noise resulting in a distorted signal 18 ($\{\tilde{s}_i(t)\}$) at the receiver 20. In the receiver 20, a demodulator 22 converts the signal with any distortion incurred during transmission into a demodulated bit stream 24 (Z_1, Z_2, \dots, Z_n). The demodulated bit stream 24 is then decoded by a decoder 26 which identifies and corrects errors in the "as received" message to produce a corrected, decoded output message stream 28 (m_1, m_2, \dots, m_n). The output message is passed to an information sink 30 for consumption.

[0016] Several channel coding techniques are used for FEC in communication systems. One technique, which is thought to achieve performance near the theoretical or Shannon limit of the transmission channel, is turbo coding. Turbo codes are interleaved, parallel concatenated, codes. The constituent codes are often convolutional codes but may be codes produced by other techniques.

[0017] Referring to FIG. 2, an exemplary turbo-code encoder 40 (indicated by a bracket) comprises two parallel, recursive, systematic, convolutional, constituent encoders 42 and 44 (indicated by brackets) with an interleaver 46 attached to the input of one of the encoders 44. Turbo code encoders may have more than two parallel constituent encoders with additional interleaver attached to the inputs of the additional constituent encoders. The turbo code encoder 40 has three intermediate outputs (c_0) 47, (c_1) 48, and (c_2) 50 which are combined into a coded serial bit stream (C) 52 by an output multiplexer 54. The code rate of the turbo code encoder 40 or the ratio of the number of input message symbols (m) to the number of output code symbols (C) is 1/3.

[0018] The turbo code encoder 40 is a systematic encoder with the symbols of the input message or frame (m) 56 constituting one of encoder's intermediate outputs (c_0) 47. The intermediate outputs (c_1) 48 and (c_2) 50 are the outputs of the first 42 and second 44 constituent convolutional encoders, respectively. The first 42 and second 44 constituent convolutional encoders each comprise a shift register 58 and a pair of modulo-2 adders 60 and 62. The shift registers 58 of the illustrated encoders have two individual registers, (T1) 64 and (T2) 66. The symbols of an input bit stream for the constituent encoders 42 and 44, (m) 56 or ($m(\alpha)$) 68, respectively, are shifted one bit at a time into the shift register 58. To ensure that the contents of the shift registers 58 are known at the start of the message and that all of the message bits are shifted completely through each shift register 58, the turbo code encoder includes a padding unit 70 to add sufficient bits (usually zeroes) to "flush" the shift register to a known state at the end of the message or frame. A convolutional encoder adds redundant bits to the message on a continuous basis so that the entire message is one code. However, the coded message data is often segmented into fixed length frames for transmission.

[0019] In the turbo code encoder 40 the input of the first constituent convolutional encoder 42 is the exemplary message sequence (m) 56 [10101]. In the recursive convolutional encoders 42 and 44, the modulo-2 sum of the input bit and the contents of the two registers (T1) 64 and (T2) 66 is computed in a modulo-2 adder 60 to create a feedback bit 72 (F). The output of the constituent convolutional encoder, (c_1) 48 or (c_2) 50, is the modulo-2 sum of the feedback bit 72 (F) and the contents of the T2 register 66 computed in the adder 62. The output of the first constituent convolutional encoder (c_1) 42 is the coded bit stream: [1011].

[0020] The input ($m(\alpha)$) 68 of the second constituent convolutional encoder 44 is the message sequence (m) 56 as rearranged by an interleaver 46. Interleaving rearranges the input symbols in some pseudo-random manner so that the multiple code symbols representing each input symbol will be separated from each other in the output bit stream (C) 52. The error correcting performance turbo codes is due, in part, to the use of large turbo interleavers that interleave hundreds bits. Interleaving results in two or more (depending on the number of constituent codes) independently coded symbol streams that refer to the same message bit stream. Turbo code decoders exploit this structure of the code to derive extrinsic information about bits in one symbol stream from the bits in the remaining stream(s) and use this information to estimate the correctly decoded symbols. The estimates are refined during successive iterations of the decoding process.

[0021] In the exemplary interleaver 46 the input symbols are rearranged according to a table where the first bit of the input sequence (m_1) becomes the fourth bit ($m(\alpha)_4$) of the convolutional encoder input sequence 68; the second input bit (m_2) becomes the third bit ($m(\alpha)_3$) and so forth. As a result the output (c_2) 50 of the second convolutional encoder 44 is different from that of the first constituent encoder 42 even though the encoding process within the encoder is the same. Further, as a result of interleaving, the three coded output symbols representing any input message symbol are separated in the output bit stream 52.

[0022] A convolutional encoder is a state machine and its operation can be represented by a path traced through a code tree or a trellis diagram. FIG. 3 is a trellis diagram illustrating an exemplary operation of the constituent recursive

convolutional encoder 42 of FIG. 2. Each node 80 of the trellis corresponds to a state of the shift register of the encoder 42 (listed vertically to the left of the trellis) at an input time or coding stage 82 (t_0, t_1, \dots) indicated horizontally across the top of the trellis). Two paths exit each node of the trellis. The path to be followed to the next state at the next stage of encoding is determined by the message bit causing the state transition. An input message bit (1) causes the encoder to follow the path represented by a dashed line and an input message bit (0) causes the encoder to follow the path represented by a solid line. An output code symbol 84 for each path is shown adjacent to the path. For example, at t_1 , a bit (1) is shifted into the encoder 42 causing the encoder to transition from the initial state (00) 86 (as a result of padding) to the next state (10) 88. When the encoder exits the first node (00) 86 for the next state (10) 88, it exits on the path corresponding to the input bit (1) (dashed line) which causes the encoder to output a code symbol (11) 90. Likewise, the next message bit (0) determines the exit path from the second node (state 10 at t_1) 88 producing the next code symbol (01) 92 and causing the encoder to move to its next state (11 at t_2) 94. It is desirable for the encoder to start and end in an all zero state. For an input sequence [101] two extra bits [01] are necessary to produce a zero ending state. Following the path through the trellis for an input message sequence [101] plus two padding bits [01], produces the output code: [1101100111]. The output of the encoder of FIG. 2 is the output constituent encoders (the trellis diagrams for each of the two encoders) which is multiplexed with the original message symbols.

[0023] Turbo code decoding is an iterative process and the turbo code decoder is modular in nature with the output of a first module (the result of a first iteration) being the part of the input to a first decoder of the next modular decoder for use in the next iteration of the process. Referring to FIG. 4, an exemplary modular turbo code decoder includes a number of, serially connected, constituent decoders 100 and 102 equal to the number constituent encoders in the particular turbo code encoder. Other arrangements of the decoder are possible if the number of constituent codes is greater than two, but the present invention may be utilized with these other arrangements, as well. The "as received" coded transmission (Z) 104, including any errors incurred in transmission, is received by the decoder from the demodulator 22. In an input demultiplexer 106, the "as received" data stream (Z) 104 is separated into three constituent streams: (z_0) 110, (z_{C1}) 108, and (z_{C2}) 112 representing the "as received" versions of the three intermediate outputs of the encoder (c_0) 47, (c_{C1}) 48, and (c_{C2}) 50.

[0024] In the constituent decoders 100 and 102, the soft output Viterbi algorithm (SOVA) is used to retrace the "most likely" path followed by the encoder through the trellis diagram when it produced the encoded message. Referring to FIG. 5, in applying the Viterbi algorithm the decoder operates serially through the stages. At each stage, a path metric 170 is computed for paths from the initial state to all possible states for that stage. The path metric 170 is the accumulated difference metric 172 along each path from the decoder's initial state 174 to each node 176 at the particular stage. The difference metric 172 for a path between two nodes represents a measure of the difference between a code word of the received message and the corresponding code word which the encoder would have generated in making the transition between those two nodes or states. According to the Viterbi algorithm, the "most likely" path followed by the encoder in coding the sequence is the path with the least path metric. This path is retained for use at the next input stage and the path with the greater path metric is discarded. As the algorithm is applied for input stages further and further into the trellis, a single "most likely" path emerges after a delay 180. There is a high probability that errors in the received message will be eliminated and that the message will be correctly decoded by following this path through the trellis diagram.

[0025] A modified Viterbi algorithm (the soft output Viterbi algorithm or SOVA) is used in turbo code decoders to produce "soft" or analog outputs from soft or hard inputs. "Soft" inputs and outputs quantize the level of the actual signal (often at eight levels) between the possible binary values of the signal. The "soft" value represents the reliability of the signal's value as well as its polarity.

[0026] There are two basic steps in the SOVA algorithm used in the constituent decoders of the turbo code decoder. The first or Viterbi step is similar to the Viterbi algorithm with the addition of the computation of the maximum path metric differences for each node at each stage of the trellis. In the second or update step, a window 178 is established corresponding to the delay of the Viterbi algorithm in converging on a single path. The window 178 slides along the trellis as the decoder moves from stage to stage. The minimum path metric difference is found from among all possible paths in this window that compete with the survivor path and would have led to a different bit decision than the survivor path. This minimum path metric difference is given a sign based on the hard value of the current bit and is the soft value of the SOVA output. The output of a constituent convolutional decoder applying the SOVA algorithm comprises a signed number for each decoded bit of the message. The sign of the number (the sign of the "log likelihood ratio") represents the value or polarity of the bit. The magnitude of the number represents the reliability of its proper decoding (the magnitude of the "log likelihood ratio").

[0027] With the SOVA algorithm, the path metric computations of the Viterbi algorithm are augmented by the addition of a term which incorporates extrinsic information ($L_{\text{in}}(t)$) supplied by the previous constituent decoder. For each state (k) at a particular t in the trellis, the path metric $M_{k,t}$ is computed by extending the path metrics from all states (k') at time $t-1$, with valid transitions to a state k at time t , as follows:

$$M_{k,t} = k \left\{ M_{k,t-1} + \frac{1}{2} \sum_{v=1}^N L c_{y_v,t} x_{v,t} x_{v,k} + \frac{1}{2} L_{in}(t) \right\}$$

where α = the fading amplitude of the channel; and E_s/N_0 = signal to noise ratio for the channel.

20 [0028] Referring to FIG. 4, the first constituent decoder 100 applies the SOVA algorithm to the systematic data (Z_0) 110, the coded data from the first encoder (Z_{C1}) 108, and initial a-priori information ($L_{in1,i}$) 114 to compute a soft output ($L(sov1)$) 116 comprising a plurality of signed numbers each representing the value of a decoded message bit and the probability that the value has been correctly decoded. The extrinsic information ($L(e)$) 120 is produced by subtracting the product of systematic data (Z_0) 110 and the channel reliability factor (L_c) 124 produced in the multiplier 122 and the a-priori information ($L_{in1,i}$) 114 from the decoder output ($L(sov1)$) 116 in a subtraction unit 118. After interleaving the extrinsic information ($L(e)$) 120 becomes a-priori information ($L_{in2,i}$) 126 which is supplied to the second constituent decoder 102.

[0029] The input data for the second constituent decoder 102 comprises the interleaved version of a-priori information (Lin2,2) 126 from the first constituent decoder 100, the systematic data (Z_0) 110 which has also been interleaved in the interleaver 128, and the "as received" coded output of the second encoder of the turbo code encoder (Z_{C2}) 112 which has been separated from the input message bit stream; (Z) 104 by the input demultiplexer 106. The second constituent decoder 102 applies the SOVA algorithm to this second coded information and produces a second soft output (Lsova2) 130, completing an iteration of the decoding process.

35 [0030] If the required number of iterations have been completed, the soft output of the second decoder (Ls0va2) 130 is diverted by switch 132 to the deinterleaver 134 and passed to an conversion unit 136 where "hard" (binary) values are selected for the decoded message 138 from the "soft" (analog) values of the decoder output.

[0031] If the required number of iterations has not been completed, the interleaved intrinsic information (Lin2,j) and the interleaved systematic data (Z₀) 110 from the input to the second decoder 102 is subtracted from the soft output of the second decoder (Lsova2) 130 in the subtraction unit 140. The resulting extrinsic information (Le) 142 is an interleaved version of the intrinsic information (Lin1,j+1) 144. After deinterleaving in the deinterleaver 134, this intrinsic information (Lin1,j+1) 144 is fed back through the switch 146 to the input to the first constituent decoder 100 for the next successive iteration of the decoding process.

[0032] The output of the SOVA algorithm depends upon the value of the minimum of the maximum path metric differences in the update window along the survivor path. For low signal to noise ratios (SNR) the maximum path metric differences are likely to be small. For high SNR, the path metric differences are likely to be large resulting in large values of the SOVA output (L_{sova}). These large values have a negative effect on the convergence of the algorithm and the performance of the algorithm can be improved by limiting the output (L_{sova}).

50 **[0033]** The present inventor came to unexpected realization that the convergence of the SOVA algorithm can be monitored by comparing the number of outputs of the constituent decoders that are saturating at a limiting level at each iteration and discontinuing iteration when this number does not change from iteration to iteration. Further, the decision to continue iterating can be made by comparing the change in the number of saturating outputs produced by each of the individual constituent decoders in a single iteration.

[0034] In the present invention, the outputs (L(sova)) are directed from the constituent decoders 100 and 102 to a comparator 150 which compares each output to a threshold or limit (T). The numbers of L(sova) outputs from the second constituent decoder 102 (L(sova2)) for a first iteration (N2,i) and a successive iteration (N2,j+1) and from the first constituent decoder 100 for the successive iteration (N1,j+1) equaling the threshold (T) are counted by the counter 151 and stored in memory 152, 154, and 156. The three numbers of saturated L(sova) values are compared in a decision unit 158. If the three numbers are approximately equal, the algorithm can be declared to have converged and the oper-

ation of the turbo code decoder can be terminated 162 without loss of information. If the three numbers are not equal, the algorithm is continuing to converge and the turbo code decoder will be allowed to continue 164 to operate unless the maximum number of iterations 160 has been reached. A second technique and apparatus stores and compares the numbers of outputs of the first and second constituent decoders which have saturated in successive iterations. When the numbers of saturating outputs do not in succeeding iteration does not change convergence is declared and operation of the decoder is halted. Likewise, the saturating outputs of a single constituent decoder can monitored for successive iterations until there is little or no further change in the numbers of saturating outputs. A fourth, less complex technique is to compare the output of the last serial constituent decoder with the output of the first serial constituent decoder performing one iteration. While the convergence of the algorithm is indicated by a lack of change in the number of saturated outputs produced by the first and last decoders performing a single iteration, there is less certainty of the effect of continuing operation using this method. If all of the values in the SOVA outputs of the constituent decoders have saturated the algorithm has converged and operation may be terminated. The threshold (T) and the L(sova) value are both signed values. The number of comparisons may be reduced by 50% by comparing outputs of only one sign with threshold having the same sign.

[0035] The terms and expressions that have been employed in the foregoing specification are used as terms of description and not of limitation, and there is no intention, in the use of such terms and expressions, of excluding equivalents of the features shown and described or portions thereof, it being recognized that the scope of the invention is defined and limited only by the claims that follow.

- 20 21. Claims**
- 1.** A method of optimizing the performance of an iterating turbo code decoder including a plurality of constituent decoders (100; 102) comprising the steps of:
- 25 (a)** establishing a limit for an output of a said constituent decoder (100; 102);
- (b)** determining a number of said outputs approximately equaling said limit for each of successive said iterations by said turbo code decoder; and
- (c)** terminating operation of said turbo code decoder when said numbers of said outputs are substantially unchanged for successive iterations;
- 30 2.** The method of claim 1 wherein said number of said outputs approximately equaling said limit is a number of said outputs of one sign approximately equaling said limit having said same sign;
- 3.** The method of claim 1 wherein a said constituent decoder (100; 102) applies the soft output Viterbi algorithm;
- 35 4.** A method of optimizing the performance of an iterating turbo code decoder including a plurality of serial constituent decoders (100, 102) comprising the steps of:
- (a)** establishing a limit for an output of a said constituent decoder (100, 102);
- 40 (b)** determining a first and a second number of said outputs approximately equaling said limit produced by a first and a second said serial constituent decoder (102) while performing a first iteration and a second iteration; respectively;
- (c)** determining a third number of said outputs approximately equaling said limit produced by a first said serial constituent decoder (100) while performing said second iteration; and
- 45 (d)** terminating operation of said turbo code decoder when said first, second, and third numbers of said outputs are substantially equal;
- 5.** The method of claim 4 wherein said first, second, and third numbers of said outputs are numbers of said outputs of one sign approximately equaling said limit having said same sign;
- 50 6.** The method of claim 4 wherein a said constituent decoder (100; 102) applies the soft output Viterbi algorithm;
- 7.** A method of optimizing the performance of an iterating turbo code decoder including a plurality of serial constituent decoders comprising the steps of:
- 55 (a)** establishing a limit for an output of a said constituent decoder (100; 102); and
- (b)** terminating operation of said turbo code decoder when substantially all said outputs produced by said constituent decoder (100; 102) while performing a said iteration are approximately equal to said limit;

8. The method of claim 7 wherein said substantially all said outputs are substantially all said outputs of one sign approximately equaling said limit of said same sign.
9. The method of claim 7 wherein a said constituent decoder (100, 102) applies the soft output Viterbi algorithm.
10. A method of optimizing the performance of an iterating turbo code decoder including a plurality of serial constituent decoders (100, 102) comprising:
 - (a) establishing a limit for an output of a said constituent decoder (100, 102);
 - (b) determining a first number of said outputs approximately equaling said limit produced by a first said serial constituent decoder (100) performing an iteration;
 - (c) determining a second number of said outputs approximately equaling said limit produced by a last said serial constituent decoder (102) performing said iteration; and
 - (d) terminating operation of said turbo code decoder when said first and second numbers of said outputs are substantially equal.
11. The method of claim 10 wherein said first and said second numbers of said outputs are numbers of said outputs of one sign approximately equaling said limit having said same sign.
12. The method of claim 10 wherein a said constituent decoder (100, 102) applies the soft output Viterbi algorithm.
13. An iterating turbo code decoder comprising:
 - (a) a plurality of serial constituent decoders (100, 102),
 - (b) a comparator (150) to compare a plurality of outputs of a said constituent decoder (100, 102) to a threshold value for said outputs;
 - (c) a counter (151) to count a number of said outputs produced by said constituent decoder (100, 102) approximately equaling said threshold value; and
 - (d) a decision unit (158) to terminate operation of said turbo code decoder when said number of said outputs approximately equaling said threshold value is unchanged for successive iterations.
14. The turbo code decoder of claim 13 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.
15. An iterating turbo code decoder comprising:
 - (a) a plurality of serial constituent decoders (100, 102),
 - (b) a comparator (150) to compare an output of a said constituent decoder (100, 102) to a threshold value for said output,
 - (c) a counter (151) to count a first and a second number of said outputs produced, respectively, by a first said serial constituent decoder (100) and a subsequent said serial constituent decoder (102) approximately equaling said threshold value; and
 - (d) a decision unit (158) to terminate the operation of said turbo code decoder when said first number of said outputs is approximately equal to said second numbers of said outputs.
16. The turbo code decoder of claim 15 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.
17. An iterating turbo code decoder comprising:
 - (a) a plurality of serial constituent decoders (100, 102),
 - (b) a comparator (150) to compare an output of a said constituent decoder (100, 102) to a threshold value for said output,
 - (c) a counter to count a number of said outputs approximately equaling said threshold value;
 - (d) a memory (152, 154, 156) to store a first, a second, and a third said number of said outputs approximately equaling said threshold value, said outputs produced, respectively, by a last said serial constituent decoder (102) performing a first and a subsequent iteration and a first said serial constituent decoder (100) performing said subsequent iteration; and

(e) a decision unit (158) to terminate the operation of said turbo code decoder when said first, second, and third numbers of said outputs are approximately equal;

18: The turbo code decoder of claim 17 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

19: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

20: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

21: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

22: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

23: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

24: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

25: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

26: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

27: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

28: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

29: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

30: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

31: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

32: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

33: The turbo code decoder of claim 18 wherein said constituent decoder (100, 102) is a soft output Viterbi algorithm decoder.

FIG. 1

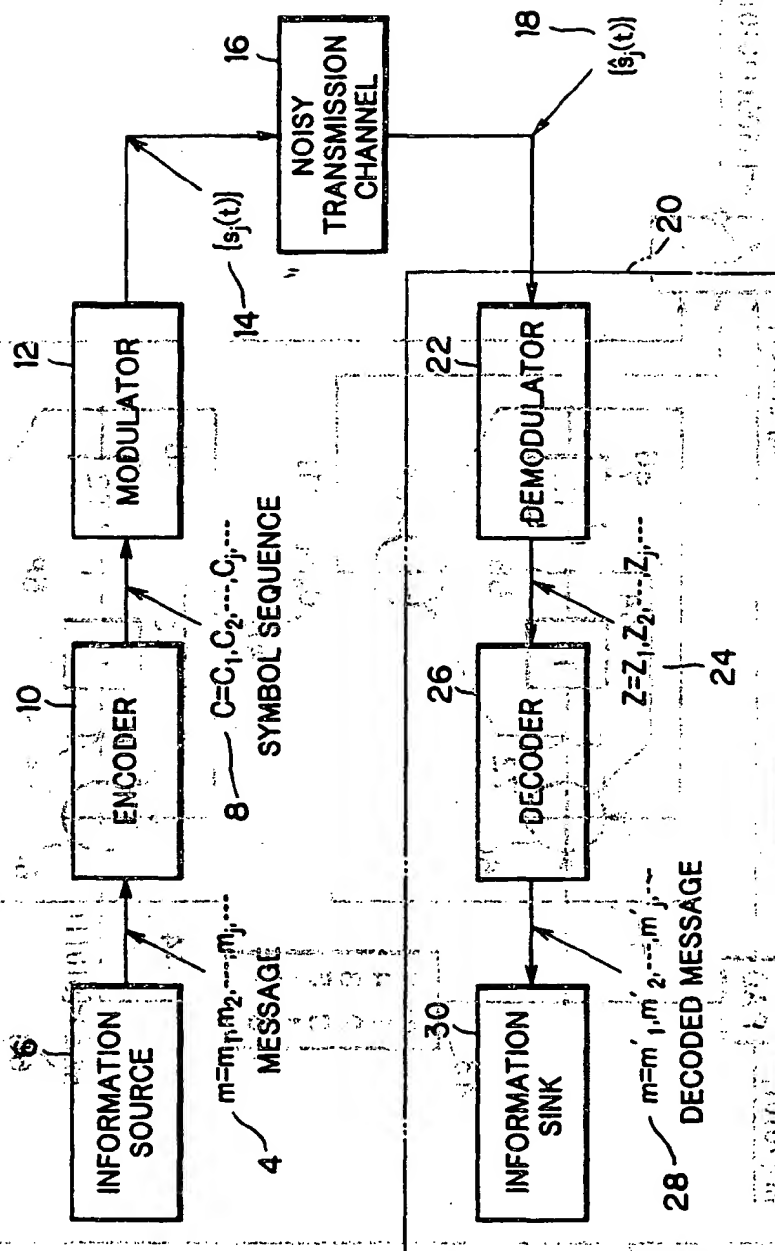
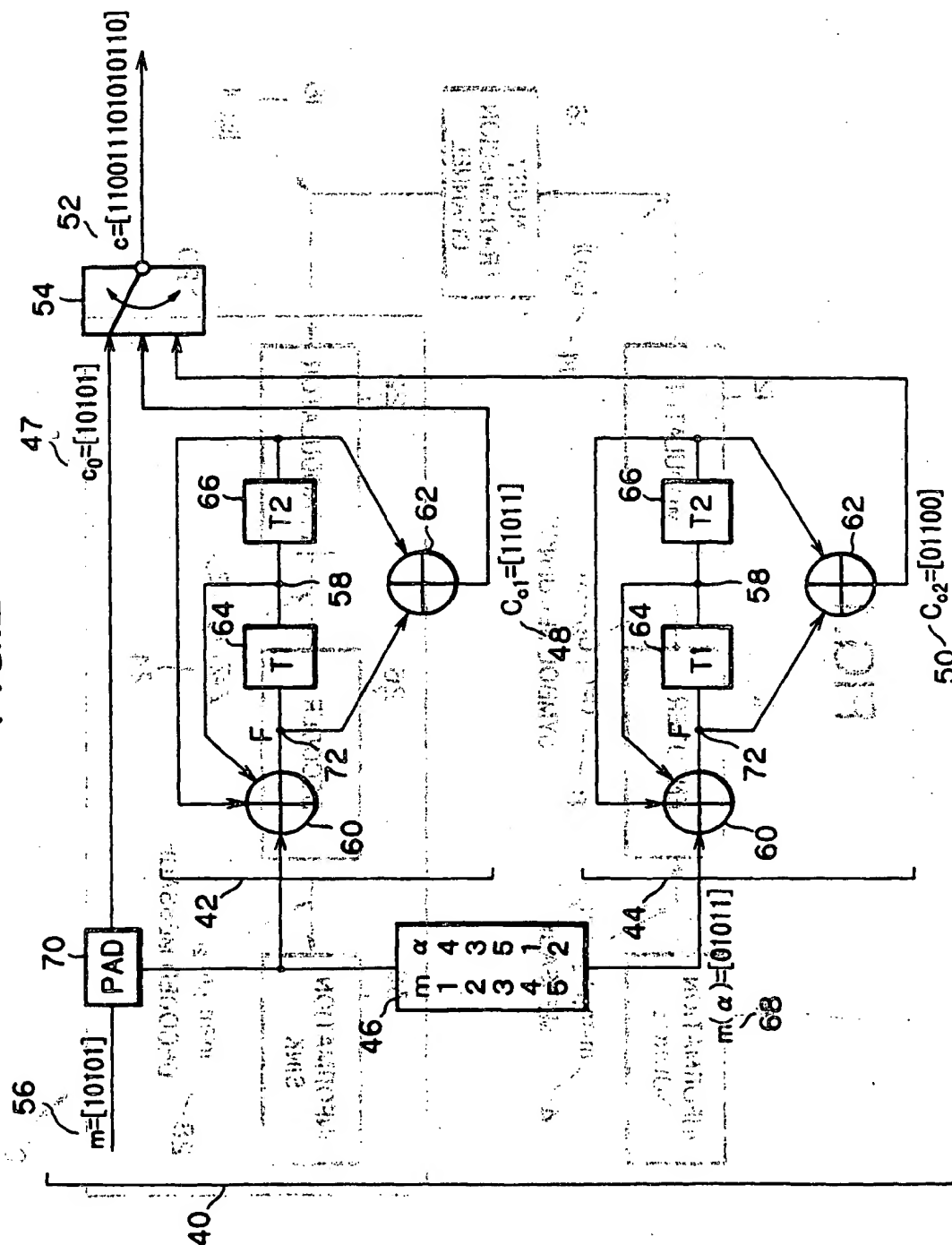


FIG.2



361

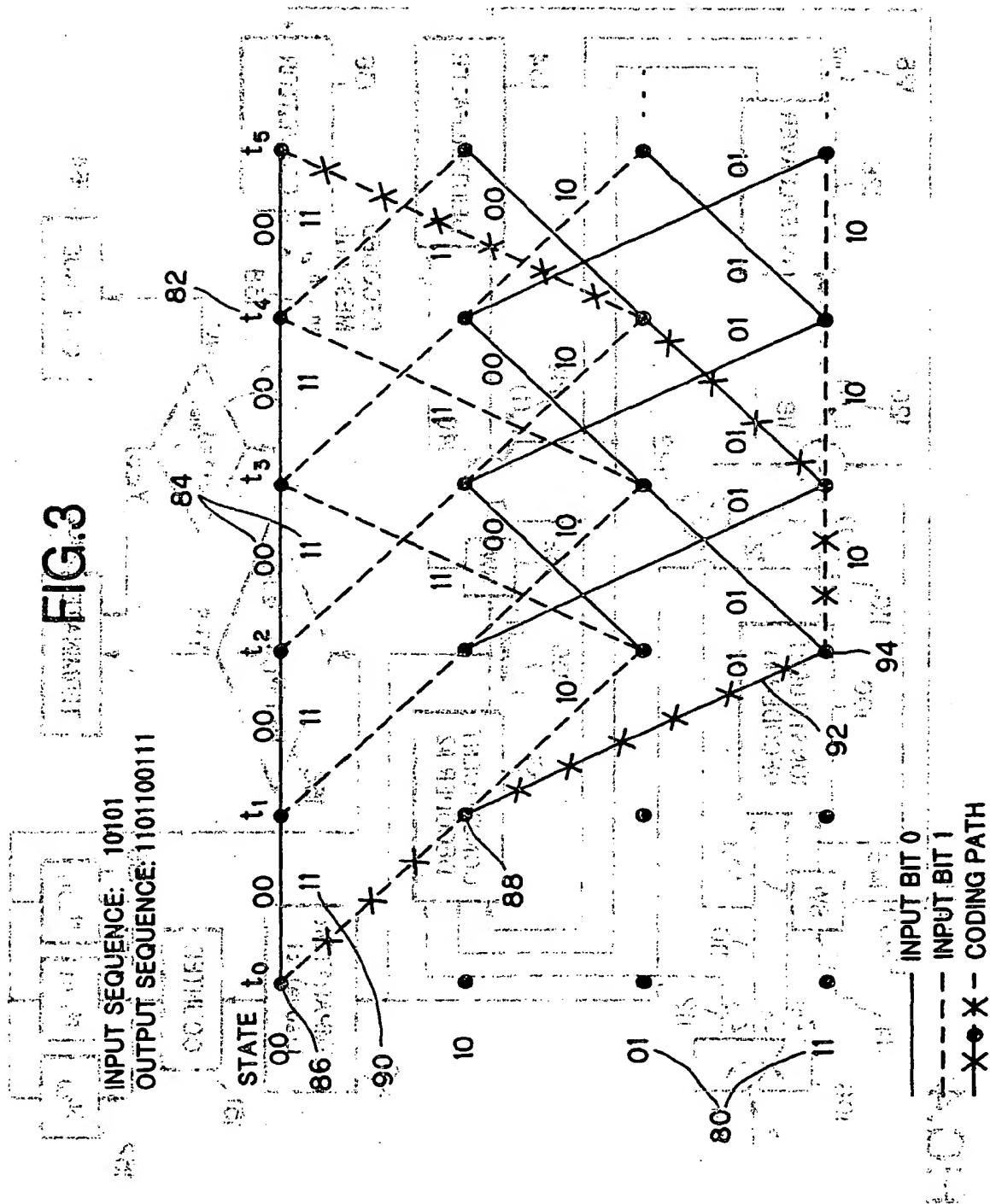


FIG. 4

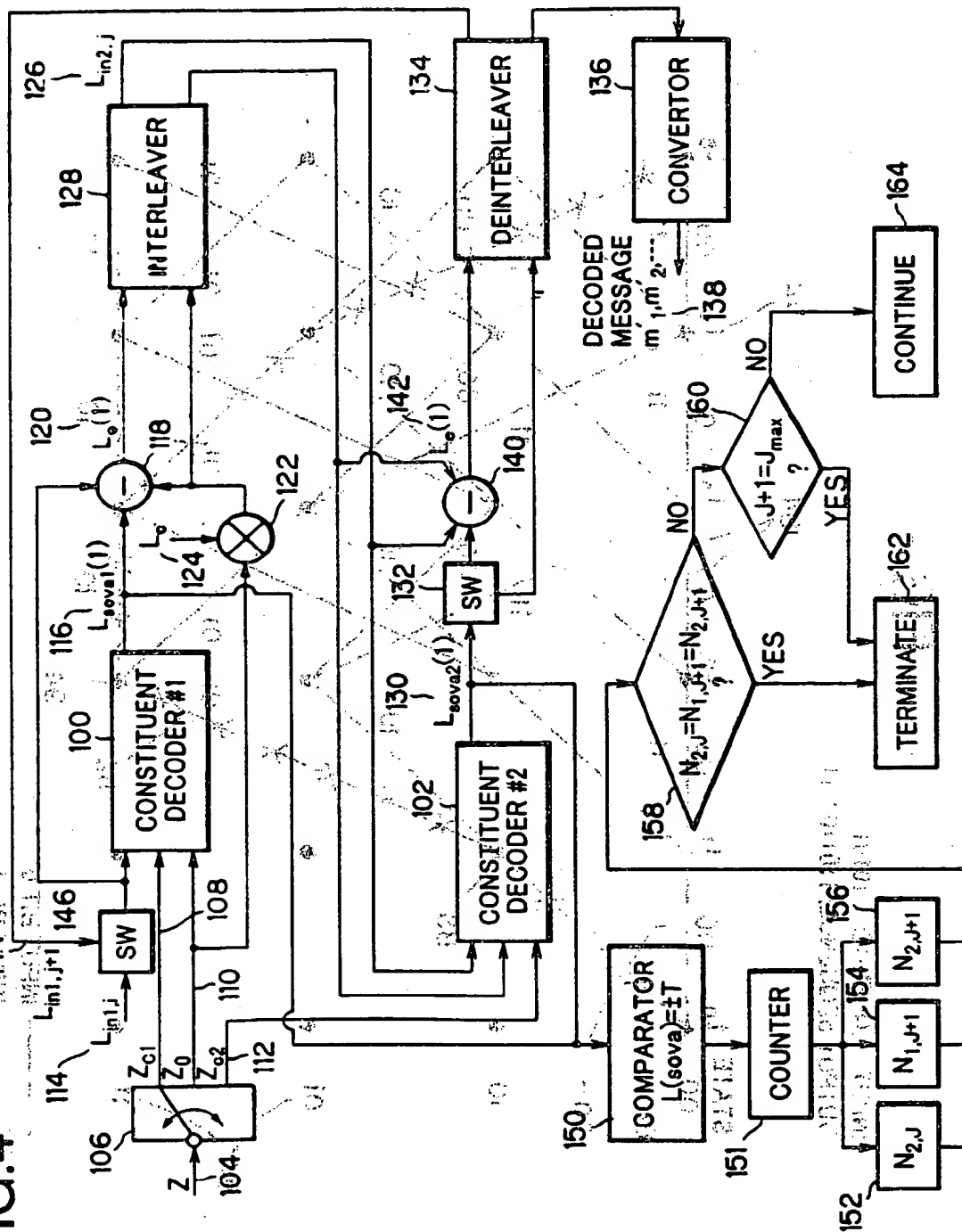
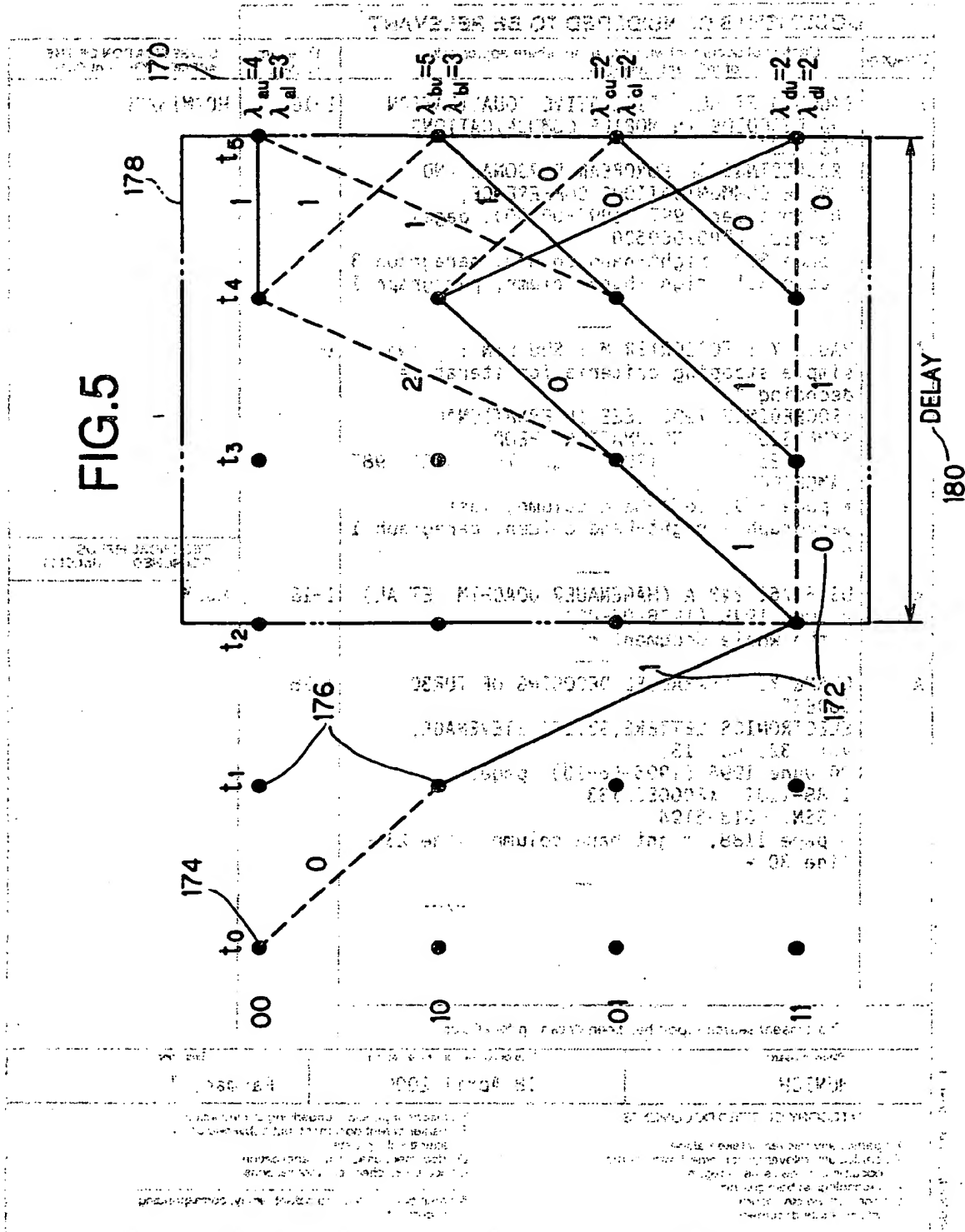


FIG.5





European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 00 10 1174

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	BAUCH G ET AL: "ITERATIVE EQUALIZATION AND DECODING IN MOBILE COMMUNICATIONS SYSTEMS" PROCEEDINGS OF EUROPEAN PERSONAL AND MOBILE COMMUNICATIONS CONFERENCE, 30 September 1997 (1997-09-30), pages 308-312, XP002060630 * page 310, right-hand column, paragraph 3 - page 311, right-hand column, paragraph 1 *	1-18	H03M13/29
A	HAO R Y ; FOSSORIER M ; SHU LIN : "Two simple stopping criteria for iterative decoding" PROCEEDINGS 1998 IEEE INTERNATIONAL SYMPOSIUM ON INFORMATION THEORY, 16 - 21 August 1998, page 279 XP002135987 CAMBRIDGE, MA, USA * page 279, left-hand column, last paragraph - right-hand column, paragraph 1 *	1-18	
A	US 5 761 248 A (HAGENAUER JOACHIM ET AL) 2 June 1998 (1998-06-02) * the whole document *	1-18	H03M
A	CHANG Y: "PARALLEL DECODING OF TURBO CODES" ELECTRONICS LETTERS, GB, IEE STEVENAGE, vol. 32, no. 13, 20 June 1996 (1996-06-20), pages 1188-1189, XP000599183 ISSN: 0013-5194 * page 1188, right-hand column, line 23 - line 30 *	1-18	
The present search report has been drawn up for all claims			
Place of search MUNICH		Date of completion of the search 18 April 2000	Examiner Farman, T
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document</p>			

EPO FORM 1503 03.82 (P04C01)



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 00 10 1174

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	CHANG Y: "Q-ary Turbo Codes with QAM Modulations" PROCEEDINGS OF ICUPC - 5TH INTERNATIONAL CONFERENCE ON UNIVERSAL PERSONAL COMMUNICATIONS, CAMBRIDGE, MA, USA, vol. 2, 29 September 1996 (1996-09-29) - 2 October 1996 (1996-10-02), pages 814-817. XP002135988 * page 815, paragraph IV *	1-18	
P, A	SHAO R Y ET AL: "TWO SIMPLE STOPPING CRITERIA FOR TURBO DECODING" IEEE TRANSACTIONS ON COMMUNICATIONS, US, IEEE INC. NEW YORK, vol. 47, no. 8, August 1999 (1999-08), pages 1117-1120. XP000848102 ISSN: 0090-6778	1-18	
			TECHNICAL FIELDS SEARCHED (Int.Cl.7)
The present search report has been drawn up for all claims			
Place of search MUNICH		Date of completion of the search 18 April 2000	Examiner Farman, T
CATEGORY OF CITED DOCUMENTS			
X: particularly relevant if taken alone Y: particularly relevant if combined with another document of the same category A: technological background O: non-written disclosure P: intermediate document		T: theory or principle underlying the invention E: earlier patent document, but published on, or after the filing date D: document cited in the application L: document cited for other reasons &: member of the same patent family, corresponding document	

EPO FORM 1503 03 82 (P/AC01)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 00 10 1174

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on 18-04-2000. The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

18-04-2000

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5761248	A	02-06-1998	19526416 A: 23-01-1997
		0755122 A: 22-01-1997	
<p>For more details about this annex see Official Journal of the European Patent Office, No. 12/82</p>			

EPO FORM P0459

reduced output

1. The first 1000 words of the text are the most important.

1. *Chlorophyll a* (Chl *a*) is the primary photosynthetic pigment in most plants and algae. It is a green pigment that absorbs light energy in the blue and red regions of the visible spectrum.



1946-1947-1948-1949-1950-1951-1952-1953-1954-1955-1956-1957-1958-1959-1960-1961-1962-1963-1964-1965-1966-1967-1968-1969-1970-1971-1972-1973-1974-1975-1976-1977-1978-1979-1980-1981-1982-1983-1984-1985-1986-1987-1988-1989-1990-1991-1992-1993-1994-1995-1996-1997-1998-1999-2000-2001-2002-2003-2004-2005-2006-2007-2008-2009-2010-2011-2012-2013-2014-2015-2016-2017-2018-2019-2020-2021-2022-2023-2024-2025-2026-2027-2028-2029-2030-2031-2032-2033-2034-2035-2036-2037-2038-2039-2040-2041-2042-2043-2044-2045-2046-2047-2048-2049-2050-2051-2052-2053-2054-2055-2056-2057-2058-2059-2060-2061-2062-2063-2064-2065-2066-2067-2068-2069-2070-2071-2072-2073-2074-2075-2076-2077-2078-2079-2080-2081-2082-2083-2084-2085-2086-2087-2088-2089-2090-2091-2092-2093-2094-2095-2096-2097-2098-2099-2100-2101-2102-2103-2104-2105-2106-2107-2108-2109-2110-2111-2112-2113-2114-2115-2116-2117-2118-2119-2120-2121-2122-2123-2124-2125-2126-2127-2128-2129-2130-2131-2132-2133-2134-2135-2136-2137-2138-2139-2140-2141-2142-2143-2144-2145-2146-2147-2148-2149-2150-2151-2152-2153-2154-2155-2156-2157-2158-2159-2160-2161-2162-2163-2164-2165-2166-2167-2168-2169-2170-2171-2172-2173-2174-2175-2176-2177-2178-2179-2180-2181-2182-2183-2184-2185-2186-2187-2188-2189-2190-2191-2192-2193-2194-2195-2196-2197-2198-2199-2200-2201-2202-2203-2204-2205-2206-2207-2208-2209-2210-2211-2212-2213-2214-2215-2216-2217-2218-2219-2220-2221-2222-2223-2224-2225-2226-2227-2228-2229-2230-2231-2232-2233-2234-2235-2236-2237-2238-2239-2240-2241-2242-2243-2244-2245-2246-2247-2248-2249-2250-2251-2252-2253-2254-2255-2256-2257-2258-2259-2260-2261-2262-2263-2264-2265-2266-2267-2268-2269-2270-2271-2272-2273-2274-2275-2276-2277-2278-2279-2280-2281-2282-2283-2284-2285-2286-2287-2288-2289-2290-2291-2292-2293-2294-2295-2296-2297-2298-2299-2300-2301-2302-2303-2304-2305-2306-2307-2308-2309-2310-2311-2312-2313-2314-2315-2316-2317-2318-2319-2320-2321-2322-2323-2324-2325-2326-2327-2328-2329-2330-2331-2332-2333-2334-2335-2336-2337-2338-2339-2340-2341-2342-2343-2344-2345-2346-2347-2348-2349-2350-2351-2352-2353-2354-2355-2356-2357-2358-2359-2360-2361-2362-2363-2364-2365-2366-2367-2368-2369-2370-2371-2372-2373-2374-2375-2376-2377-2378-2379-2380-2381-2382-2383-2384-2385-2386-2387-2388-2389-2390-2391-2392-2393-2394-2395-2396-2397-2398-2399-2400-2401-2402-2403-2404-2405-2406-2407-2408-2409-2410-2411-2412-2413-2414-2415-2416-2417-2418-2419-2420-2421-2422-2423-2424-2425-2426-2427-2428-2429-2430-2431-2432-2433-2434-2435-2436-2437-2438-2439-2440-2441-2442-2443-2444-2445-2446-2447-2448-2449-2450-2451-2452-2453-2454-2455-2456-2457-2458-2459-2460-2461-2462-2463-2464-2465-2466-2467-2468-2469-2470-2471-2472-2473-2474-2475-2476-2477-2478-2479-2480-2481-2482-2483-2484-2485-2486-2487-2488-2489-2490-2491-2492-2493-2494-2495-2496-2497-2498-2499-2500-2501-2502-2503-2504-2505-2506-2507-2508-2509-2510-2511-2512-2513-2514-2515-2516-2517-2518-2519-2520-2521-2522-2523-2524-2525-2526-2527-2528-2529-2530-2531-2532-2533-2534-2535-2536-2537-2538-2539-2540-2541-2542-2543-2544-2545-2546-2547-2548-2549-2550-2551-2552-2553-2554-2555-2556-2557-2558-2559-2560-2561-2562-2563-2564-2565-2566-2567-2568-2569-2570-2571-2572-2573-2574-2575-2576-2577-2578-2579-2580-2581-2582-2583-2584-2585-2586-2587-2588-2589-2590-2591-2592-2593-2594-2595-2596-2597-2598-2599-2600-2601-2602-2603-2604-2605-2606-2607-2608-2609-2610-2611-2612-2613-2614-2615-2616-2617-2618-2619-2620-2621-2622-2623-2624-2625-2626-2627-2628-2629-2630-2631-2632-2633-2634-2635-2636-2637-2638-2639-2640-2641-2642-2643-2644-2645-2646-2647-2648-2649-2650-2651-2652-2653-2654-2655-2656-2657-2658-2659-2660-2661-2662-2663-2664-2665-2666-2667-2668-2669-2670-2671-2672-2673-2674-2675-2676-2677-2678-2679-2680-2681-2682-2683-2684-2685-2686-2687-2688-2689-2690-2691-2692-2693-2694-2695-2696-2697-2698-2699-2700-2701-2702-2703-2704-2705-2706-2707-2708-2709-2710-2711-2712-2713-2714-2715-2716-2717-2718-2719-2720-2721-2722-2723-2724-2725-2726-2727-2728-2729-2730-2731-2732-2733-2734-2735-2736-2737-2738-2739-2740-2741-2742-2743-2744-2745-2746-2747-2748-2749-2750-2751-2752-2753-2754-2755-2756-2757-2758-2759-2760-2761-2762-2763-2764

[illegible]

1. *Chlorophyll a* and *Chlorophyll b* were determined by the method of Arar and Collins (1971) using a Shimadzu 1010 spectrophotometer.

1. The above information is true and correct to the best of my knowledge and belief. I have signed this statement in the presence of the undersigned witnesses, who are not related to me by blood or marriage, and who are not interested in the outcome of the proceedings in this case.

the 1990s, the number of people in the United States who are 65 years of age or older is projected to increase from 20 million to 30 million, and the number of people 75 years of age or older is projected to increase from 10 million to 15 million (U.S. Census Bureau, 1996). The number of people 85 years of age or older is projected to increase from 2 million to 4 million (U.S. Census Bureau, 1996). The number of people 90 years of age or older is projected to increase from 500,000 to 1 million (U.S. Census Bureau, 1996). The number of people 95 years of age or older is projected to increase from 100,000 to 200,000 (U.S. Census Bureau, 1996). The number of people 100 years of age or older is projected to increase from 10,000 to 20,000 (U.S. Census Bureau, 1996).

1. The first step is to identify the problem or question that needs to be addressed. This involves understanding the context and the specific requirements of the task.

DATE: 11/06/2008 TIME: 10:29

This Page Blank (ushto)

This Page Blank (copy)

1. The first part of the document is a letter from the President of the United States to the Congress, dated January 1, 1861. It is a copy of the original letter, and is signed by Abraham Lincoln.

On the other hand, the fact that the β values are not significantly different from zero indicates that the model is not misspecified.

[illegible]

STANDARD 2011 10/10/11 10:10:11 AM 10/10/11 10:10:11 AM

1. The first step is to identify the problem or question that needs to be answered. This involves understanding the context and the specific requirements of the task.

There are no other persons who have been identified as having been involved in the activities of the TCM or its affiliates.

and the other "factors" are "fixed" and "given."

and government officials. On 19 June 1997, the National Government of the Philippines issued a statement that the Philippines was not involved in the 1995-1996 military operations in East Timor. The statement also stated that the Philippines was not involved in the 1995-1996 military operations in East Timor.

1. The first step is to identify the problem or question that needs to be answered. This involves understanding the context and the specific requirements of the task.

1. $\lim_{x \rightarrow 0} \frac{1}{x} = \infty$ (for $x > 0$)

1993, 1994, 1995

1. The first step in the process is to identify the problem or issue that needs to be addressed. This involves gathering information and understanding the context of the situation.

[illegible]

1. The first step is to identify the problem. This involves understanding the current situation and the goals that need to be achieved.

1. The first step is to identify the problem or question that needs to be answered. This involves understanding the context and the specific requirements of the task.

... and the price of business is not $\frac{1}{2}$ the $\frac{1}{2}$ the price of the goods. ...

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840.

[illegible]

... ..

1. *Pharmaceuticals* (1997) 10, 11.

the β -phase of the polymer. The β -phase is the more stable phase at low temperatures and is the phase that is observed in the solid state. The α -phase is the more stable phase at high temperatures and is the phase that is observed in the liquid state. The transition from the β -phase to the α -phase is a first-order phase transition and is characterized by a change in the heat capacity and the volume of the polymer.

1. The first step is to identify the problem or question that needs to be answered. This involves understanding the context and the specific requirements of the task.

[illegible]

44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

This Page Blank (uspto)